稀疏条件下的重叠子空间聚类算法 *

邱云飞 [a]，费博雯 [b†]，刘大千 [c]，刘　兴 [a]

(辽宁工程技术大学 a. 软件学院; b. 工商管理学院; c. 电子与信息工程学院，辽宁 葫芦岛 125105)

摘　要：现有子空间聚类算法不能很好地平衡子空间数据的稠密性和不同子空间数据稀疏性的关系，且无法处理数据的重叠问题。针对上述问题，提出一种稀疏条件下的重叠子空间聚类(OSCSC)算法。算法利用 l_1 范数和 Frobenius 范数的混合范数表示方法建立子空间表示模型，并对 l_1 范数正则项进行加权处理，提高不同子空间的稀疏性和同一子空间的稠密性；然后对划分好的子空间使用一种服从指数族分布的重叠概率模型进行二次校验，判断不同子空间数据的重叠情况，进一步提高聚类的准确率。在人造数据集和真实数据集上分别进行测试，实验结果表明，OSCSC 算法能够获得良好的聚类结果。

关键词：重叠子空间聚类；混合范数；重叠概率模型；指数族分布

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.08.0904

Novel algorithm of overlapping subspace clustering under sparse condition

Qiu Yunfei[a], Fei Bowen[b†], Liu Daqian[c], Liu Xing[a]

(*a. School of Software, b. School of Business & Management, c. School of Electronics Information Engineering Liaoning Technical University, Huludao Liaoning* 125105, *China*)

Abstract: The existing subspace clustering algorithms cannot balance the density of the data in the same subspace and the sparsity of the data between different subspaces and most algorithms cannot solve the overlap of data. To solve the above problems, this paper proposed a novel algorithm of overlapping subspace clustering algorithm under sparse condition (OSCSC) . The algorithm used the mixed norm representation method of L1 norm and Frobenius norm to establish the subspace representation model, and the weighted L1 norm regular term could improve the sparsity of different subspaces and the density of the same subspace. Then, the algorithm performed rechecks on the partitioned subspaces by using an overlapping probability model subject to exponential family distribution to determine whether exist overlapping in different subspaces, which could further improve the accuracy of clustering. The results of the experiment on both artificial datasets and real-world datasets show that the algorithm has better clustering performance by being compared to other contrast algorithms.

Key Words: overlapping subspace clustering; mixed norm; overlapping probability model; exponential family distribution

0　引言

聚类分析是数据挖掘领域的重要研究内容之一，在机器学习、医学生物分析和计算机视觉等方面具有广泛应用[1~3]。近些年来数据规模迅速增长，数据规模和维度也越来越大，在处理和分析这样的数据集时，由于样本分布稀疏，数据间距离几乎相同，传统的聚类方法往往无法获得准确的聚类结果[4]。

为了解决数据规模较大和数据维度较高等问题，Agrawal 等人[5]首次将子空间聚类的概念应用于聚类问题的分析中。此后，根据这一思想，国内外学者和研究人员相继提出了许多子空间聚类方法。现有子空间聚类方法可大致被分为 5 类：迭代方法、代数方法、统计方法、基于矩阵分解的方法和基于谱聚类的方法[6]。其中较为流行的子空间聚类算法是基于谱聚类框架的方法，例如稀疏子空间聚类(sparse subspace clustering, SSC)[7]方法、最小二乘回归子空间聚类算法(least squares regression, LSR)[8]和低秩表示子空间聚类算法(low-rank representation, LRR)[9]。以上三种方法通过将某个数据样本由其他样本线性表示构建相似度矩阵并将其转换为 Laplacian 矩阵，然后对该矩阵特征分解，根据谱聚类的思想对得到的特征值和特征向量进行聚类。在此基础上，Xu 等人[10]提出重加权稀疏子空间聚类(reweighted sparse subspace clustering, RSSC)方法，利用 log-sum 启发方法对 l_1 范数进行迭代加权，提高子空间稀疏

基金项目：国家自然科学基金青年科学基金资助项目（61401185）

作者简介：邱云飞（1976-），男，辽宁阜新人，教授，博士，主要研究方向为数据挖掘、机器学习；费博雯（1991-），女（通信作者），辽宁抚顺人，博士研究生，主要研究方向为数据挖掘、机器学习（feibowen2098@163.com）；刘大千（1992-），男，辽宁铁岭人，博士研究生，主要研究方向为图像与视觉信息计算、目标检测与跟踪；刘兴（1995-），男，辽宁沈阳人，硕士研究生，主要研究方向为数据挖掘．

性。Zhang 等人[11]提出一种基于低秩表示的子空间聚类改进算法，在构建子空间聚类模型时，采用核范数和 Frobenius 范数代替秩函数，并利用非精确的增广拉格朗日乘子方法进行优化得到最优系数矩阵。

上述子空间聚类算法虽然从一定程度上提高了聚类性能，但由于其均属于硬划分聚类方法，忽略了数据簇间存在重叠的问题。在实际处理子空间聚类问题时，不同子空间的数据会存在重叠部分，不完全独立，在进行聚类时会导致一些数据不能归到正确的子空间内，影响聚类精度[12]。根据数据划分不确定性，Banerjee 等人[13]提出一种基于模型的重叠聚类算法(model-based overlapping clustering, MOC)，该算法利用概率模型的方法处理数据间的重叠问题，判断某一数据样本是否属于多个类簇。Fu[14]等人提出了一种贝叶斯重叠子空间聚类算法(Bayesian overlapping subspace clustering, BOSC)，通过构建数据矩阵的层次生成模型来发现数据间的重叠结构，该算法没有充分考虑数据间的结构关系，在处理维数较高的数据集时效率较低。

针对上述聚类算法存在的问题，本文提出一种稀疏条件下的重叠子空间聚类算法(overlapping subspace clustering under sparse condition,OSCSC)，算法利用加权 l_1 范数和 Frobenius 范数的混合范数子空间表示方法对数据空间进行划分，能够更好地保证同一子空间数据的稠密性和不同子空间数据的稀疏性。然后使用一种重叠概率模型对子空间内的数据进行重叠校验，判断其是否属于多个子空间。本文算法主要基于稀疏子空间聚类方法的基础上对数据样本进行重叠判断，允许一个数据对象属于多个子空间，当子空间划分过程发生错误时有利于将数据归于到正确的子空间内，能够有效提高聚类的准确率。

1 子空间聚类

近年来，子空间聚类成为处理高维数据的主要研究方法。其中以谱聚类为基础的子空间聚类方法得到了广泛关注。该类方法通过寻找低维空间的表示系数构造基于自表达模型的相似矩阵，然后利用谱聚类算法得到最终的聚类结果。SSC 和 LSR 算法均具有这样的性质，认为在整个数据空间内的每个数据点 x_j 可用其他数据点线性表示

$$x_j=\sum_{i\neq j}^{n} z_{ij}x_i+\eta_j \tag{1}$$

其中：n 为数据点个数；z_{ij} 为数据点 x_i 和 x_j 之间的相似系数，用于构建相似矩阵；η_j 用于描述自表示模型中的偏差或噪声数据。当 x_i 和 x_j 不属于同一子空间时 $z_{ij}=0$。

SSC 算法求解子空间聚类模型可表示为

$$\min_z \|Z\|_1 \quad \text{s.t.} \quad X=XZ\ ,\ z_{jj}=0 \tag{2}$$

其中：$\|\cdot\|_1$ 为 l_1 范数，$X=[x_1,\cdots,x_N]$ 为数据矩阵。Z 是由 z_j 组成的系数矩阵。对于有噪声数据，SSC 可以扩展为

$$\min_z\|X-XZ\|_F^2+\|Z\|_1 \quad s.t.\ z_{jj}=0 \tag{3}$$

LSR 通过最小化系数矩阵 Z 的 Frobenius 范数建立如下目标函数：

$$\min_z \|Z\|_F^2 \quad \text{s.t.} \quad X=XZ\ ,\ z_{jj}=0 \tag{4}$$

对于有噪声数据，LSR 可扩展为

$$\min_z\|X-XZ\|_F^2+\lambda\|Z\|_F^2 \quad s.t.\ z_{jj}=0 \tag{5}$$

根据相似矩阵的对称性和非负性，相似矩阵 W 可被定义为

$$U=\frac{1}{2}(|Z|+|Z^{\mathrm{T}}|) \tag{6}$$

然后将谱聚类应用于相似矩阵中获得聚类结果。

2 稀疏条件下的重叠子空间聚类

本文提出的 OSCSC 算法融合子空间聚类以及重叠聚类的思想，旨在不同子空间的重叠聚类问题。本文算法利用迭代加权的 l_1 范数和 Frobenius 范数的混合范数表示方法建立子空间聚类模型，将高维数据通过低维子空间线性表示，使用线性交替方向法对模型进行优化。然后对得到了子空间结果采用重叠概率模型估计数据的重叠情况。本文算法与以往基于硬划分的子空间聚类技术的不同之处在于允许某一数据属于一个或多个子空间，可进一步提高聚类精度，减少聚类错误。

2.1 加权混合范数的子空间表示

2.1.1 子空间表示模型

子空间聚类方法核心在于子空间模型的建立。本文结合 SSC 和 LSR 算法思想，提出一种混合范数的子空间表示方法，在保证类间数据稀疏的同时，增加类内数据的稠密性。定义 X 为 $M\times N$ 的数据矩阵，包含 N 个列向量 $\{x_j\in \mathrm{R}^M\}_{j=1}^N$ 可由 J 个子空间线性表示，子空间聚类的目的是将数据矩阵 X 中的列向量 x_j 划分到正确的子空间中。子空间聚类模型可表示为

$$\min_z\ \lambda\|Z\|_1+\frac{1-\lambda}{2}\|Z\|_F^2 \tag{9}$$

对于有噪声的数据集，模型可以表示为

$$\min_z\ \lambda\|Z\|_1+\frac{1-\lambda}{2}\|Z\|_F^2+\|X-XZ\|_F^2 \tag{10}$$

其中：$\|\cdot\|_F$ 为 Frobenius 范数；$\lambda\in[0,1]$ 为权衡系数，用于权衡两个正则化项间的关系；Z 为系数矩阵，提供了关于子空间的向量分割问题的条件。

2.1.2 加权方式

在处理实际的问题时，真实数据集内的数据间情况复杂，只通过 l_1 正则化项保证子空间稀疏性的效果并不理想。文献[10,15]均提出针对 l_1 正则化项的加权方法，并经过大量实验证明迭代更新权值的 l_1 范数($\min_{x\in R^n}\|wx\|_1$)与单独运用 l_1 范数($\min_{x\in R^n}\|x\|_1$)相比能够获得更为稀疏的系数结构，平衡了 l_1 范数和 l_0 范数间的差异，使 l_1 更能逼近 l_0。文献[14]通过求解

$$\min_{x\in R^n}\sum_{i=1}^{n}\ln\left(|x_i|+\varepsilon\right)\ ,\qquad \text{s.t.}\quad Ax=b \tag{11}$$

迭代更新得到加权方式

$$w_i^{(t)}=\left(\left|x_i^{(t-1)}\right|+\varepsilon\right)^{-1} \tag{12}$$

其中：w_i^t 为数据点 x_i 在第 t 次迭代中对应的权值，ε 为控制参数。本文将迭代加权 l_1 范数的思想应用于混合范数子空间表示模型中，则式(10)可以表示为

$$\min_{z}\ \lambda\|W\odot Z\|_1+\frac{1-\lambda}{2}\|Z\|_F^2+\|X-XZ\|_F^2 \tag{13}$$

2.1.3 模型优化求解

针对优化问题式(13)可将其转换为

$$\min_{z}\ \lambda\|W\odot Z\|_1+\frac{1-\lambda}{2}\|Z\|_F^2+\|E\|_F^2 \quad \text{s.t.}\ \ X-XZ=E\ ,\ z_{jj}=0 \tag{14}$$

对上述问题利用线性交替方向法(linearized alternating directions method, LADM)[16]优化。通过引入拉格朗日乘子 μ，得到增广拉格朗日函数

$$L(Z,E,\mu)=\lambda\|W\odot Z\|_1+\frac{1-\lambda}{2}\|Z\|_F^2+\|E\|_F^2 +\frac{\rho}{2}\|X-XZ-E\|_F^2+\text{tr}(\mu^{\text{T}}(X-XZ-E))\quad \text{s.t.}\ z_{jj}=0 \tag{15}$$

其中：$\rho>0$ 为惩罚参数。利用 LADM 优化问题式(15)，首先定义 k 为迭代次数，固定 $E^{(k)}$，更新 $Z^{(k+1)}$，即

$$Z^{(k+1)}=\min_{Z^{(k)}}L(Z^{(k)},E^{(k)},\mu^{(k)})\approx S_{\frac{\lambda}{\theta\mu^{(k)}}}\left\{Z^{(k)}+\frac{1}{\theta}[X^{\text{T}}(X-XZ^{(k)}-E^{(k)}+\frac{\mu^{(k)}}{\rho^{(k)}})-\frac{1-\lambda}{2\rho^{(k)}}Z^{(k)}]\right\}\quad \text{s.t.}\ z_{jj}^{(k+1)}=0 \tag{16}$$

其中：$S_\tau(q)$ 为收缩阈值算子，具体定义为 $S_\tau(q)=\text{sgn}(q)\max(|q|-\tau,0)$，$\theta=1.1\sigma^2$（$\sigma$ 为数据矩阵 X 的最大奇异值）。

然后固定 $Z^{(k+1)}$，更新 $E^{(k+1)}$，即

$$E^{(k+1)}=\min_{Z^{(k)}}L(Z^{(k+1)},E^{(k)},\mu^{(k)})=\frac{\mu^{(k)}+\rho^{(k)}(X-XZ^{(k+1)})}{2+\rho^{(k)}}\quad \text{s.t.}\ z_{jj}^{(k+1)}=0 \tag{17}$$

最后，分别更新拉格朗日乘子 μ 和惩罚参数 ρ，LADM 具体优化过程如下。

算法 1 利用 LADM 优化问题式(15)

输入： 数据矩阵 X，权衡系数 λ

输出： 系数矩阵 Z，噪声矩阵 E

初始化：$Z^{(0)}=0$，$E^{(0)}=0$，$\mu^{(0)}=0$，$\rho^{(0)}=10^{-6}$，$\gamma=1.6$，$\theta=1.1\sigma^2$，$\varepsilon_1=10^{-4}$，$\varepsilon_2=10^{-5}$

固定 $E^{(k)}$，更新 $Z^{(k+1)}$

$$Z^{(k+1)}=\min_{Z^{(k)}}L(Z^{(k)},E^{(k)},\mu^{(k)})\approx S_{\frac{\lambda}{\theta\mu^{(k)}}}\left\{Z^{(k)}+\frac{1}{\theta}[X^{\text{T}}(X-XZ^{(k)}-E^{(k)}+\frac{\mu^{(k)}}{\rho^{(k)}})-\frac{1-\lambda}{2\rho^{(k)}}Z^{(k)}]\right\}\quad \text{s.t.}\ z_{jj}^{(k+1)}=0$$

固定 $Z^{(k+1)}$，更新 $E^{(k+1)}$

$$E^{(k+1)}=\min_{Z^{(k)}}L(Z^{(k+1)},E^{(k)},\mu^{(k)})=\frac{\mu^{(k)}+\rho^{(k)}(X-XZ^{(k+1)})}{2+\rho^{(k)}}\quad \text{s.t.}\ z_{jj}^{(k+1)}=0$$

更新权值

$$w_i^{(k+1)}=\frac{1}{\left(\left|z_i^{(k)}\right|+\varepsilon\right)}$$

更新拉格朗日乘子

$$\mu^{(k+1)}=\mu^{(k)}+\rho^{(k)}(X-XZ^{(k+1)}-E^{(k+1)})$$

更新惩罚参数 $\rho^{(k+1)}=\min(\gamma\cdot\rho^{(k)},10^{10})$

收敛条件：$\left\|X-XZ^{(k)}-E^{(k)}\right\|_F/\|X\|_F<\varepsilon_1$，

$$\max(\left\|E^{(k)}-E^{(k-1)}\right\|_F/\|X\|_F,\left\|Z^{(k)}-Z^{(k-1)}\right\|_F/\|X\|_F)<\varepsilon_2,$$

由上式可得出优化后的系数矩阵 Z^*，进而得到相似矩阵 $U=\frac{1}{2}(|Z^*|+|(Z^*)^{\text{T}}|)$，然后利用一种标准分割法 Ncut[17]对子空间进行分割并得到子空间集合。

2.2 重叠概率模型

尽管使用加权的混合范数子空间表示方法能够提高同一子空间数据的稠密性和不同子空间数据的稀疏性，但子空间聚类过程中依然存在错误，且该方法属于硬划分聚类方法，一般只允许一个数据样本仅属于一类，当聚类发生错误时无法校验。针对此问题，本文使用一种重叠概率模型用于判断已划分的子空间内数据是否可以属于多个子空间，给定高维数据集 X，将已得到 L 个子空间集合表示为 $\bigcup_{l=1}^{L}S_l$，其中每个子空间都代表一类数据。设 $Y=Y_1\cup Y_2\cup\cdots\cup Y_L$ 为来自子空间 $\bigcup_{l=1}^{L}S_l$ 的数据。针对 (S,Y) 进行二次校验，判断子空间是否存在重叠部分。这样当数据被误分时可以及时校验其是否属于其他子空间，能够进一步提高聚类的准确率。

重叠概率模型是一种服从指数分布族[18]的概率模型，指数分布族的定义是指概率分布满足以下形式

$$f_Y(y\,|\,\theta)=exp\{\theta T(y)-\varphi(\theta)\} \tag{18}$$

其中：$T(y)$ 是分布的充分统计量，θ 为自然参数，$\varphi(\theta)$ 为累积量函数。

不同子空间内重叠概率模型的条件概率可表示为

$$p(y\,|\,\theta)=\sum_{b}\frac{\pi(b)}{c(b)}\prod_{l=1}^{L}p(y_i\,|\,\theta_l)^{b_l} \tag{19}$$

其中：$b=[b_1,\cdots,b_L]$ 为一组布尔向量(潜变量)，用于判断数据的重叠情况，$b_l\in\{0,1\}$，b 中每个元素均对应一个子空间。$c(b)$ 为归一化数。定义 $\pi(b)$ 为 b 的先验，每个向量 b 都对应着一个 $\pi(b)$，若向量 b 中所有元素均为零时，则令 $\pi(0)=0$，表明在数据集内会存在一些离值点，它们并不属于任何簇，但是这些值往往不能被忽略不计，由于其不属于重叠概率模型的结构，所以式(19)可以表示为

$$p(y|\theta)=\begin{cases}\sum_b \frac{\pi(b)}{c(b)}\prod_{l=1}^{L}p(y_i|\theta_l)^{b_l}, & b\neq 0\\ p(y|\theta_{l+1}), & b=0\end{cases} \tag{20}$$

重叠概率模型的每个 component 都服从于同一指数族分布，则式(19)可被改写为

$$p(y|\theta,b)=\frac{1}{c(b)}\exp\left\{\sum_{l=1}^{L}b_l\theta_l T(y)-b_l\varphi(\theta_l)\right\} \tag{21}$$

直接计算得

$$c(b)=\exp\left\{\varphi\left(\sum_{l=1}^{L}b_l\theta_l\right)-\sum_{l=1}^{L}b_l\varphi(\theta_l)\right\} \tag{22}$$

利用 $c(b)$ 的闭合形式可以得到

$$p(y|\theta,b)=\exp\left\{T(y)\sum_{l=1}^{L}b_l\theta_l-\varphi\left(\sum_{l=1}^{L}b_l\theta_l\right)\right\} \tag{23}$$

从上式可以看出条件概率 $p(y|\theta,b)$ 的每项 component 均服从自然参数为 $\sum_{l=1}^{L}b_l\theta_l$ 的指数族分布。

由于潜变量 b 是一组布尔向量，$\pi(b)$ 为 b 的先验，其中每个元素服从伯努利分布 Bernoulli ϕ_l，而 ϕ_l 先验分布服从贝塔分布 Beta$\{\alpha_l,\beta_l\}$。将其代入式(21)中可以得到概率模型的联合分布为

$$p(y,b,\phi|\alpha,\beta,\theta)=\frac{1}{c(b)}\left(\prod_{l=1}^{L}p(\phi_l|\alpha_l,\beta_l)p(b_l|\phi_l)\right)\times\left(\prod_{l=1}^{L+1}p(y|\theta_l)^{b_l}\right) \tag{24}$$

重叠子空间聚类的目的是判断高维数据的不同子空间的数据簇是否存在重叠关系。通过将高维数据利用低维子空间线性表示，得到数据点较为相似且稠密的子空间，针对这些子空间使用重叠概率模型用于判断数据簇间的重叠问题。判断过程中需要对模型的参数进行估计，还要对每个数据 y_i 推断出潜在簇的赋值向量 b。对于每个数据点 (y_i,b) 是条件独立的，所以重叠概率模型可通过求解条件概率的最大化似然函数求得，需要通过对联合概率函数乘积取 log 并将其最大化，可表示为

$$L(b,\alpha,\beta,\theta)=\sum_{i=1}^{n}\log p(y_i,b_i|\alpha,\beta,\theta) \tag{25}$$

根据式(24)可得到

$$\begin{aligned}&\max\sum_{i=1}^{n}\log p(y_i,b_i|\alpha,\beta,\theta)=\\&\max\left[\sum_{i=1}^{n}\log p(b_i|\alpha,\beta)+\sum_{i=1}^{n}\log p(y_i|b_i,\theta)\right]=\\&\max\left[\sum_{i=1}^{n}\sum_{l=1}^{L}\log\left(\int_{\phi_{i,l}}p(b_{i,l}|\phi_{i,l})p(\phi_{i,l}|\alpha_l,\beta_l)d\phi_{i,l}\right)\right]+\\&\max\left[\sum_{i=1}^{n}\left(\sum_{l=1}^{L+1}b_{i,l}\log p(y_i|\theta_l)-\log c(b_i)\right)\right]\end{aligned} \tag{26}$$

由上式可知，计算重叠数据的关键在于二值向量 b 的选择以及参数 α,β,θ 的估计。

2.3 参数估计

在参数估计部分使用交替最大化算法[18]对子空间重叠概率模型中的参数进行估计，其过程主要分为两个部分：布尔向量 b 的选择和 α,β,θ 参数估计。在布尔向量选择部分，给定 α,β,θ 的参数值，由于每个子空间的数据点定义了一组布尔向量 $b_{i,l}$，利用 $b_{i,l}$ 优化得到最大对数似然。在参数估计部分，给定布尔向量集 $b_{i,l}$ 的参数值，利用参数 α,β,θ 的值来优化对数似然函数，具体过程如下：

1)布尔选择

a)给定初始 α,β,θ 值；

b)对于任意数据点，令 b_{i,l_0} 为初始赋值向量(简写为 b_0)，定义向量 v_l 为第 l 个元素为 1，其余元素为 0 的布尔向量，构成布尔集 $V=\{v_1,\cdots,v_L\}$；

c)将布尔向量的迭代计算亦分为 l($l=1,\ldots,L$)层，采用快速启发式迭代法[13]计算每层所选的布尔向量的最优解 $b_{i,l}$；

d)利用模拟退火算法在选择布尔向量的过程中跳出局部极值；

对于每层得到布尔向量 $b_{i,l}$ 设置一个初始温度参数 T_0，则布尔向量可表示为 $b_{i,l}^{T_0}$，定义 $f_T<1$ 为乘性因子，用于保证在每次迭代中温度参数 T 均呈下降趋势，最大迭代次数 J。在得到新的一层布尔值时需要进行迭代判断，将搜索产生的新的布尔值 $b_{i,l}^{T^*}$ 与 $b_{i,l}^{T_0}$ 进行比较，若 $b_{i,l}^{T^*}<b_{i,l}^{T_0}$ 或达到最大迭代次数时，该层迭代终止。选择每一层的最优布尔向量集 b_{l*}。(本文参数取值：T_0=50，f_T=0.67，J=40)。

当 b_{l*} 的值为 1 时，认为该样本数据属于其对应的子空间。若 $b_{l*}=0$ 则认为该样本数据不属于对应的子空间。当一组布尔向量中有两个或两个以上元素的布尔值为 1 时，可以认为该样本数据为重叠数据，可属于多个子空间。

2) α,β,θ 参数估计

a)参数 α,β 估计。给定布尔向量集 $b_{i,l}$，令 t_l 表示 $b_{i,l}$ 中 1 的个数，则 $n-t_l$ 表示 $b_{i,l}$ 中 0 的个数。最优贝塔分布参数满足下式：

$$\frac{\alpha_l}{\beta_l}=\frac{t_l}{n-t_l}$$

在本文实验中取 $\beta_l=1$，则 $\alpha_l=t_l/(n-t_l)$。

b)参数 θ 估计。利用式(26)的对数似然函数第二部分计算最优极值 θ_l^*；

计算最优极值 θ_l^* 的具体推导过程见附录 1。

2.4 算法描述

算法 2 OSCSC

输入： 数据集 X，权衡系数 λ，类数 L，初始温度 T_0。

输出： 聚类结果。

利用式(13)得到加权的 l_1 和 Frobenius 混合范数的子空间表示模型;

根据算法 1 优化得到 Z^*，并得到求出相似矩阵 $U=\frac{1}{2}(|Z|+|Z^T|)$；

使用一种标准分割方法 Ncut 对子空间进行分割并得到子

空间集合 (S,Y)；

对得到的 (S,Y) 使用重叠概率模型得到其联合分布函数；

根据式(25)得到最大似然函数 $L(b,\alpha,\beta,\theta)$ 并估计参数。

for 每次迭代

固定 (α,β,θ)，优化 b。

固定 b，使用优化 (α,β,θ)。

对于每层选取的布尔向量利用模拟退火算法搜索最优解，即得到最终的布尔向量值 b_{l^*}，并根据 b_{l^*} 判断重叠情况。

3 实验结果与分析

为了验证 OSCSC 算法的有效性，实验选取 5 种聚类算法与本文算法进行对比，并在人造数据集以及真实基准数据集测试算法性能。5 种对比算法分别为 SSC，LSR，RSSC，MOC 和 BOSC。本文在 MATLAB R2016a 编程环境下实现 。

实验采用聚类准确率 AC(accuracy)[19-20]、标准化互信息 NMI(normalized mutual information)[21]以及运行时间作为评价准则用于评价 OSCSC 算法性能。为了提高算法的可靠性，将实验中的 6 种算法独立运行 10 次，AC 计算公式为

$$AC=\frac{\sum_{i=1}^{N}\delta(s_i,map(r_i))}{N}$$

其中：N 为样本总数，$\delta(\ )$ 函数表示的意义为：当函数内两个参数相等时，函数值为 1，否则为 0。s_i 为样本数据原始类别，r_i 为聚类后样本数据的类别，$map(r_i)$ 为映射函数，将聚类后的类别映射为与样本原始类别等价的类别。

NMI 的计算公式为

$$NMI=\frac{\sum_{i=1}^{c}\sum_{j=1}^{c}n_{ij}\ln\frac{Nn_{ij}}{n_in_j}}{\sqrt{\left(\sum_{i=1}^{c}n_i\ln\frac{n_i}{N}\right)\left(\sum_{j=1}^{c}n_j\ln\frac{n_j}{N}\right)}}$$

其中：N 为样本总数，c 为类簇数，n_i 和 n_j 分别表示为属于类簇 i 和类簇 j 的样本数，n_{ij} 为属于类簇 i 和类簇 j 的相同样本的个数。

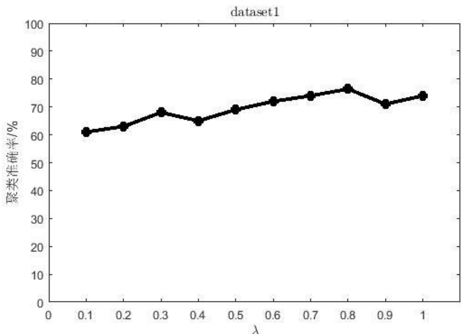
3.1 人造数据集实验结果

为了测试算法性能，本节实验采用文献[22]的方法随机生成两个人造数据集 dataset1 和 dataset2。人造数据集 dataset1 中包含 500 个数据样本，类数为 4，维度为 30。人造数据集 dataset2 中包含 3 000 个数据样本，类数为 6，维度为 80。为了更加接近真实数据集，在人造数据集中定义不同类之间存在一定的重叠样本，用于测试算法发现重叠簇的能力。人造数据集 dataset1 和 dataset2 的基本信息如表 1 所示。

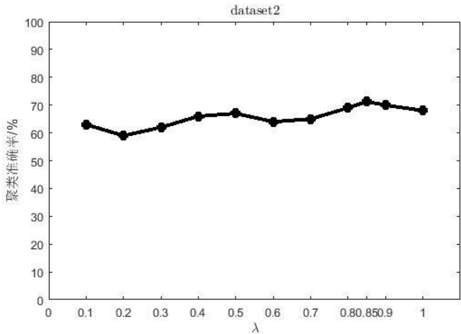
本文使用权衡系数 λ 平衡 l_1 范数和 Frobenius 范数之间的关系，通过改变 λ 的值使得聚类效果最佳。在 2 个人造数据集上 λ 与算法聚类准确率间的关系如图 1 所示，根据图 1 可知当 λ 分别取 0.8 和 0.85 时使得算法聚类准确率最佳。实验将 OSCSC 算法与其他 5 种算法分别在两个人造数据集上进行测试。6 种算法在 dataset1 上的实验结果见表 2。6 种算法在 dataset2 上的实验结果见表 3。

表 1 人造数据集信息

数据集	类别	样本数	维度
dataset1	1	62	30
	2	207	30
	3	136	30
	4	95	30
dataset2	1	253	80
	2	657	80
	3	510	80
	4	649	80
	5	389	80
	6	542	80



(a)人造数据集 1



(b)人造数据集 2

图 1 权衡系数 λ 与聚类准确率的关系

表 2 6 种算法在 dataset1 上的实验结果对比(%)

		SSC	RSSC	LSR	MOC	BOSC	OSCSC
AC	均值	68.83	72.53	71.37	68.55	72.91	76.43
	标准差	1.003	0.885	0.542	1.126	0.937	0.953
NMI	均值	69.54	70.56	70.87	69.47	74.29	77.52
	标准差	0.857	0.962	1.233	0.779	1.025	0.874
运行时间/s	均值	5.16	6.42	1.13	10.63	8.74	7.51

表 3 6 种算法在 dataset2 上的实验结果对比(%)

		SSC	RSSC	LSR	MOC	BOSC	OSCSC
AC	均值	62.87	65.53	64.74	64.23	68.96	70.35
	标准差	1.705	1.139	1.224	0.721	1.337	1.125
NMI	均值	64.05	64.81	62.57	66.42	68.32	72.14
	标准差	1.418	1.059	1.263	0.931	1.163	0.985
运行时间/s	均值	54.93	60.19	13.89	92.57	77.83	74.22

根据 dataset1 和 dataset2 数据集上的实验结果可知 MOC、

BOSC 和 OSCSC 算法的聚类准确率要优于 SSC、LSR、RSSC 等基于硬划分的聚类方法，但 MOC 和 BOSC 算法在处理样本数较大、数据分布稀疏且具有一定维度的数据集时，不能充分利用数据空间信息，导致算法运行速率较差。本文算法利用加权的混合范数子空间表示方法将高维数据空间划分为若干低维子空间集合，从一定程度上减少了对样本数据集直接进行重叠聚类的难度，能够较好地处理具有一定规模和一定维度的数据集。根据表 2 和 3 的结果可以看出本文算法在两种规模不同的人造数据集上均取得了较为理想的聚类性能。通过对比 6 种聚类算法的平均运行时间可知，其中 LSR 算法运行效率较高，OSCSC 算法由于在判断重叠数据时采用概率模型并使用模拟退火的方法使该模型尽可能收敛于全局最优解，保证聚类的准确性，这样会增加算法的运行时间，但 MOC 和 BOSC 算法相比较而言，本文算法的运行时间较为理想。

为了进一步验证 OSCSC 算法处理噪声数据的有效性，实验在两个人造数据集上加入不同比例的噪声干扰(受干扰的数据位置随机选定)，分别测试 6 种算法的聚类准确率。噪声干扰比例依次设置为 10%、20%、30%、40%和 50%，实验结果见表 4。根据表 4 的实验结果可以看出，本文算法能够处理不同程度的噪声数据集并且与其他算法相比当噪声干扰程度的增大时 OSCSC 算法的聚类准确率变化不大，受噪声影响较小。

表 4　不同噪声程度下 6 种算法的聚类准确率　/%

数据集	噪声干扰比例	SSC	RSSC	LSR	MOC	BOSC	OSCSC
database1	10%	68.72	72.27	70.82	67.73	72.17	75.86
	20%	68.07	71.39	69.41	67.19	70.93	75.43
	30%	66.56	70.83	68.27	66.01	69.54	74.22
	40%	63.38	68.34	65.97	64.75	68.31	73.59
	50%	60.88	65.36	62.71	61.78	65.82	72.46
database2	10%	62.13	64.86	64.25	63.71	68.45	70.24
	20%	61.49	64.31	63.58	63.08	67.62	69.43
	30%	60.37	62.94	62.46	62.72	66.23	68.55
	40%	58.63	60.44	61.09	61.28	64.84	67.23
	50%	55.89	57.64	59.83	59.29	63.15	65.61

3.2　真实数据集实验结果

本节实验采用国际通用的六个真实数据集测试算法性能。真实数据集基本信息如表 5 所示。其中，musk、soybean、waveform 和 pendigits 均为 UCI 数据集(http://archive.ics.uci.edu/ml/datasets.html)。USPS 为手写数字数据集(http://www.cs.nyu.edu/~roweis/data.html)，包含 0~9 共 10 种类型的数字图像，每幅图片大小为 16×16，由一个 256 维的特征向量表示，本文从每个类中随机选取 100 幅图像(共 1000 幅图像)用于测试。AR 数据集(http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html)包含了 126 人超过 4000 幅人脸图像，本文随机选取其中 80 人共 960 幅图像进行测试，利用降采样方式将每幅图像大小将至 32×24。

表 5　真实数据集基本信息

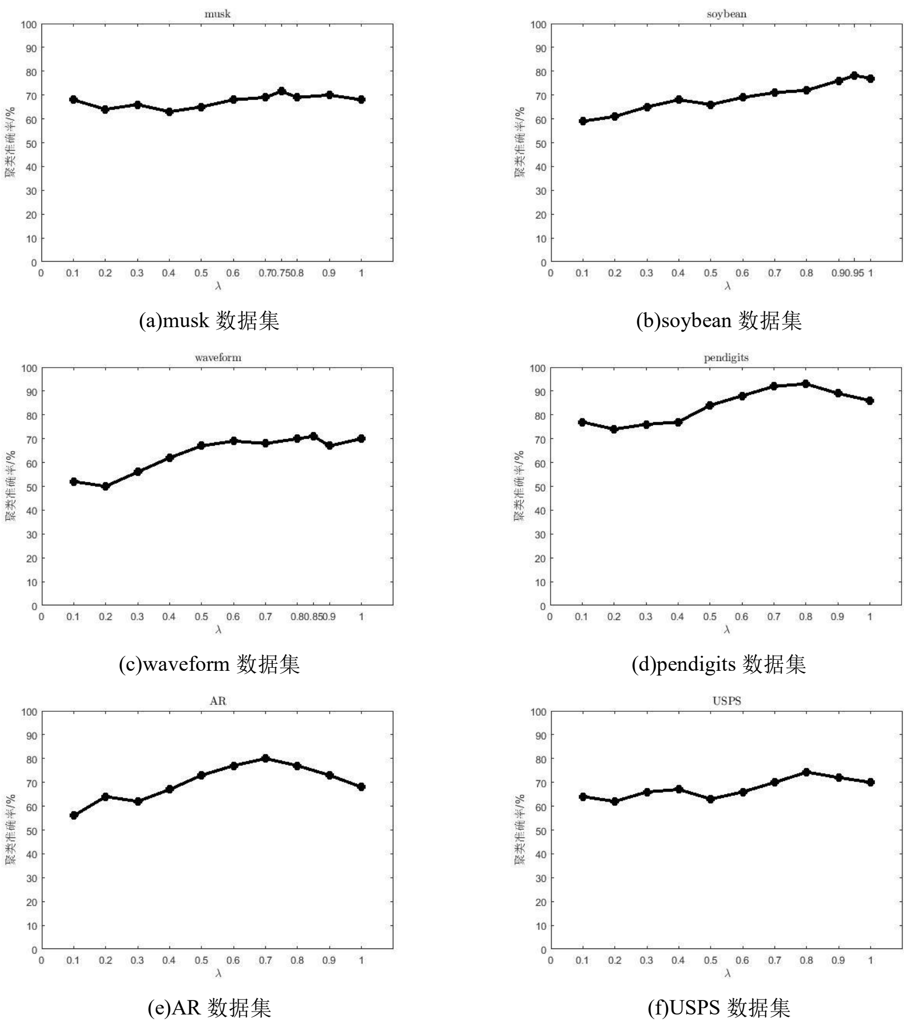
数据集	类数	样本数	维度
musk	2	476	168
soybean	19	307	35
waveform	3	5000	21
pendigits	10	10992	16
AR	80	960	768
USPS	10	1000	256

由于 λ 主要用于平衡两个正则项之间的关系，其取值会直接影响算法的聚类结果。故本节实验给出 OSCSC 算法在 6 个真实数据集上权衡系数 λ 与算法聚类准确率间的关系，如图 2 所示。由图 2 可知，在 musk 数据集上当权衡系数 $\lambda=0.75$ 时，能够得到最佳聚类结果，依次地，在 soybean、waveform、pendigits、AR 和 USPS 数据集上 λ 分别取 0.95、0.85、0.8、0.7 和 0.8 时本文算法可以获得最佳聚类结果。

将 6 种算法在六个真实数据集上进行测试，其 AC 和 NMI 的实验结果分别见表 6、7。根据表 6 和 7 的实验结果可知，OSCSC 算法在 6 个真实数据集上均能够取得较为理想的聚类结果。SSC、LSR、RSSC 算法均为基于硬划分的聚类方法，当在聚类过程中发生错误时无法及时校验，导致数据无法归于正确的类中。MOC、BOSC 和 OSCSC 算法均属于软划分方法，允许数据属于多个类别。但是 MOC 和 BOSC 在处理一定数据规模维数数据集时聚类效果不够理想。OSCSC 算法利用加权的混合范数的子空间表示方法将其分割为若干子空间，并对子空间内的数据进行聚类分析，在保证同一子空间数据的稠密性和不同子空间数据的稀疏性，尽可能将数据划分到正确子空间内。其次对得到的结果使用重叠概率模型进行二次校验，避免了在整个高维空间内直接对数据进行一一匹配，利用重叠概率模型能够有效地发现数据集内的重叠样本，将错误分配的数据归到正确的子空间中，提高了聚类的准确率。通过表 6 和 7 的每个算法结果的标准差可知，本文算法每次运行所得到的结果较为稳定。

表 6　六种算法在真实数据集上的 AC 实验结果(%)

		musk	soybean	waveform	pendigits	AR	USPS
SSC	均值	67.57	69.87	60.26	55.86	70.23	52.53
	标准差	0.824	1.197	0.735	1.434	0.768	1.005
RSSC	均值	70.37	71.54	59.87	56.93	71.52	61.67
	标准差	1.025	0.984	0.723	1.174	0.692	1.325
LSR	均值	71.51	68.11	62.72	59.43	73.68	68.71
	标准差	0.779	1.265	0.952	1.085	0.843	1.146
MOC	均值	72.32	72.71	65,22	79.28	72.41	66.43
	标准差	2.311	1.485	1.241	1.486	2.013	1.341
BOSC	均值	71.85	74.93	70.83	86.78	74.65	68.43
	标准差	0.966	0.789	0.951	0.737	0.971	0.743
OSCSC	均值	70.34	78.13	70.55	92.86	80.37	74.28
	标准差	0.836	1.147	0.972	1.336	0.624	0.858



(a)musk 数据集 (b)soybean 数据集

(c)waveform 数据集 (d)pendigits 数据集

(e)AR 数据集 (f)USPS 数据集

图 2 权衡系数 λ 与聚类准确率的关系

表 7 6 种算法在真实数据集上的 NMI 实验结果(%)

		musk	soybean	waveform	pendigits	AR	USPS
SSC	均值	58.68	60.41	53.47	50.18	64.42	53.85
	标准差	0.877	0.932	1.072	1.257	1.147	0.536
RSSC	均值	60.32	62.54	51.61	50.75	67.87	58.13
	标准差	1.042	0.848	1.105	0.836	0.773	0.834
LSR	均值	65.31	59.98	54.06	52.93	74.35	65.72
	标准差	0.948	0.964	0.937	1.375	0.887	0.641
MOC	均值	64.81	65.12	58.48	73.44	74.91	63.48
	标准差	1.258	0.924	1.263	1.575	1.429	1.126
BOSC	均值	64.13	70.29	60.62	79.94	73.83	67.28
	标准差	1.041	1.289	1.432	1.623	1.388	1.245
OSCSC	均值	64.24	73.87	61.29	86.31	78.24	72.93
	标准差	0.931	1.157	0.947	1.321	0.863	0.679

六种算法在真实数据集上的平均运行时间如表 8 所示。根据表 8 可知 LSR 算法的运行时间最短，OSCSC 算法在处理规模较大的数据集时算法的运行速率要优于与 MOC 和 BOSC 算法，这是由于本文算法在对数据样本进行重叠判断之前，充分利用高维数据的低维子空间表示结构进行子空间划分，并针对这些划分好的子空间内数据进行重叠判断，而没有从整个数据空间直接进行重叠聚类，降低了直接处理高维数据的难度，算法的运行效率与其他重叠聚类算法相比较为理想。与 SSC 和 RSSC 相比，根据不同数据集的特性而言，如果数据集本身存在重叠数据较多，本文利用重叠概率模型判断数据重叠时所时间较长，运行效率要低于 SSC、RSSC 算法。OSCSC 算法在对 pendigits 数据集进行聚类时所用时间较长，其原因是该数据集内数据的重叠情况较多，算法利用重叠概率模型处理重叠问题时将这些重叠数据分配到对应的不同子空间的过程会消耗较多的时间。其次由于 soybean 数据集和 AR 数据集内包含的类别数较多，在计算数据重叠情况时，算法需要判断的类别数较多，增加了算法的聚类时间，可见采用软划分技术的聚类算法在处理类别数较多的数据集时时间消耗过多。但综合表 6 和 7 的实验结果，本文算法的整体聚类性能要优于其他五种聚类算法。

4 结束语

本文提出的 OSCSC 算法采用加权的 l_1 范数和 Frobenius 范数的混合范数表示方法建立子空间模型，将高维数据通过低维子空间线性表示，提高了同一子空间数据的稠密性和不同子空间数据的稀疏性。对已划分的子空间使用重叠概率模型判断子

空间内数据的重叠情况，并通过交替最大化算法对模型进行参数估计，在参数估计过程中使用模拟退火的方法寻求模型的最优解，进一步将数据归于正确子空间内，提高了聚类准确率。在不同规模的人造数据集和真实数据集上的测试结果表明，OSCSC 算法能够获得较为理想的聚类结果。下一步的工作将重点研究如何提高重叠子空间算法的运行效率。

表 8　六种算法在真实数据集上的平均运行时间　/s

	musk	soybean	waveform	pendigits	AR	USPS
SSC	7.21	8.34	108.22	191.25	20.83	16.54
LSR	2.24	3.07	36.71	55.67	4.23	3.32
RSSC	9.68	11.52	110.57	207.31	28.56	21.75
MOC	24.55	45.31	146.18	294.47	68.93	63.31
BOSC	10.73	35.63	93.58	273.67	52.49	53.87
OSCSC	9.26	33.81	115.34	269.36	47.32	45.06

参考文献：

[1] Elhamifar E, Vidal R. Sparsity in unions of subspaces for classification and clustering of high-dimensional data [C]// Proc of the 49th Annual Allerton Conference on Communication, Control, and Computing. 2011: 1085-1089.

[2] Peng X, Tang H J, Zhang L, et al. A unified framework for representation-based subspace clustering of out-of-sample and large-scale Data [J]. IEEE Trans on Neural Networks and Learning Systems, 2016, 27 (12): 2499-2512.

[3] 陈爱国，王士同．基于多代表点的大规模数据模糊聚类算法 [J]．控制与决策，2016, 31 (12): 2122-2130.

[4] Hu H, Lin Z C, Feng J J, et al. Smooth representation clustering [C]// Proc of IEEE International Conference on Computer Vision and Pattern Recognition. 2014: 3834-3841.

[5] Agrawal R, Geherke J, Gunopulos D. et al. Automatic subspace clustering of high dimensional data [J]. Data Mining and Knowledge Discovering, 2005, 11 (1): 5-33.

[6] 王卫卫，李小平，冯象初，等．稀疏子空间聚类综述 [J]．自动化学报，2015, 41 (8): 1373-1384.

[7] Elhamifar E, Vidal R. Sparse subspace clustering: algorithm, theory, and applications [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2013, 35 (11): 2765-2781.

[8] Lu C Y, Min H, Zhao Z Q, et al. Robust and efficient subspace segmentation via least squares regression [C]// Proc of the 12th European Conference on Computer Vision. 2012: 347-360.

[9] Liu G C, Lin Z C, Yu Y. Robust subspace segmentation by low-rank representation [C]// Proc of the 27th International Conference on Machine Leaning. 2010: 663-670.

[10] Xu J, Xu K, Chen K, et al. Reweighted sparse subspace clustering [J]. Computer Vision and Image Understanding, 2015 138: 25-37.

[11] 张涛，唐振民，吕建勇．一种基于低秩表示的子空间聚类改进算法 [J]．电子与信息学报，2016, 38 (11): 2811-2818.

[12] Baadel S, Thabtah F, LU J. Overlapping clustering: a review [C]// Proc of IEEE Conference on SAI Computing. 2016: 233-237.

[13] Banerjee A, Krumpelman C, Ghosh J, et al. Model-based overlapping clustering [C]// Proc of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2005: 532-537.

[14] Fu Q, Banerjee A. Bayesian overlapping subspace clustering [C]// Proc of IEEE International Conference on Data Mining. 2009: 776-781.

[15] Candes E J, Wakin M B, Boyd S P. Enhancing sparsity by reweighted l1 Minimization [J]. Journal of Fourier Analysis and Applications, 2008, 14 (5): 877-905.

[16] Panagakis Y, Kotropoulos C. Elastic net clustering applied to pop//rock music structure analysis [J]. Pattern Recognition Letters, 2014, 38 (3): 46-53.

[17] Shi J B, Malik J. Normalized cuts and image segmentation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2000, 22 (8): 888-905.

[18] Fu Q, Multiplication mixture models for overlapping clustering [C]// Proc of IEEE International Conference on Data Mining. 2008: 791-796.

[19] 刘展杰，陈晓云．局部子空间聚类 [J]．自动化学报，2016, 42 (8): 1238-1247.

[20] Cai D, He X F, Han J W. Non-negative matrix factorization on manifold [C]// Proc of the 8th IEEE International Conference on Data Mining. 2008: 63-72.

[21] 邓赵红，张丹丹，蒋亦樟．基于划分自适应融合的多视角模糊聚类算法 [J]．控制与决策，2016, 31 (4): 593-600.

[22] Aggarwal C C, Procopiuc C M, Wolf J L, et al. Fast algorithms for projected clustering [C]// Proc of ACM SIGKDD International Conference on Management of Data. New York: ACM Press, 1999: 61-72.

附录 最优极值 θ_l^* 的计算

利用式(26)的对数似然函数第二部分完成参数 θ 值的估计，即

$$f(\theta)=\sum_{i=1}^{n}\left(\sum_{l=1}^{L+1}b_{i,l}\log p\left(y_i \mid \theta_l\right)-\log c\left(b_i\right)\right)$$

将式(18)代入上式可得

$$f(\theta)=\sum_{i=1}^{n}\left(T\left(y\right)\sum_{l=1}^{L+1}b_{i,l}\theta_l-\varphi\left(\sum_{l=1}^{L+1}b_{i,l}\theta_l\right)\right)$$

将上式对 θ_l 求二阶导数，可得

$$\nabla_{\theta_j}^2 f(\theta)=-\sum_{i=1}^{n}b_{i,1}\nabla_{\theta_l}^2\varphi\left(\sum_{l=1}^{L+1}b_{i,l}\theta_l\right)$$

由于 $\varphi\left(\theta\right)$ 为累积量函数，其值为正数，所以得到的 $\nabla_{\theta_l}^2 f(\theta)$ 为负，$f(\theta)$ 表现为凸函数。由此可将 $f(\theta)$ 对 θ_l 求一阶导数，使其为 0，计算最优极值 θ_l^*，其最优极值为

$$\theta_l^*=-\sum_{i=1;b_{i,l}=1}^{n}\sum_{w=1}^{L+1}b_{i,w}\theta_w+\nabla_{\theta_l}^{-1}\varphi\left(\sum_{i=1;b_{i,l}=1}^{n}T(y)\right)$$